A Formally Verified Access Control Mechanism for Information Centric Networks

Mahdi Aiash and Jonathan Loo
School of Science and Technology
Middlesex University, London, UK
{*M.Aiash, J.Loo*}*@MDX.AC.UK*

Keywords: Information-Centric Network, Network of Information, Capability-Based Access Control, Casper/FDR.

Abstract: Communications in Information-Centric Networking place more attention on WHAT data are being exchanged rather than WHO are exchanging them. A well-established approach of information centric networks is the Network of Information (NetInf) architecture, developed as part of the EU FP7 project SAIL. The security of NetInf has been fairly analysed in the literature. In particular, research efforts have been focusing on achieving data integrity and confidentially, source or publisher authenticity and authorization. This paper analyses some work in the literature to enforce authorized access to data in NetInf, highlights a potential security threat and proposes an enhancement to address the discovered threat. The new enhancement has been formally verified using formal method approach based on the Casper/FDR tool.

1 INTRODUCTION

A large number of emerging Internet applications requires information dissemination across different organizational boundaries and heterogeneous platforms. The implementation of Information Centric Networks (ICNs) (B. Ahlgren, 2012) makes efficient content distribution possible by making information retrieval host-independent and integration into the network storage for caching information. Requests for particular content can, thus, be satisfied by any host or server holding a copy. One well-established approach of ICNs is the Network of Information (NetInf) architecture (D. Kutscher, 2013), developed as part of the EU FP7 project SAIL. The approach is based on the Publish/Subscribe model, where hosts join a network, publish data, and subscribe to publications. The NetInf introduces two main stages namely, the Publication and Data Retrieval through which hosts publish and retrieve data. Also, a distributed Name Resolution System (NRS) has been introduced to map the data to its publishers. Furthermore, data objects, such as web pages, articles or videos are named and identified using the Uniform Resource Identifier for Named Information (URI-ni) format (H. Baker, 2012), hence these objects are referred to as Named Data Objects (NDOs).

The security of the NetInf architecture has been discussed and analysed in a number of research efforts in the literature such as (J. Loo, 2014) (K. Pentikousis, 2013). In NetInf architecture, security measures are defined as part of the URI-ni naming scheme (H. Baker, 2012) to guarantee the data integrity, confidentiality and data source authenticity. As described in (M. Aiash, 2014), security in NetInf focuses mainly on the security of data objects while less interest is focused on the security of the infrastructure, network elements and communication channels. Consequently, the NetInf architecture is still vulnerable to security attacks (M. Aiash, 2014) in form of data poisoning in the NSR and Denial of Service (DoS).

Therefore, the work in (M. Aiash, 2014) proposed a Registration Stage to take place prior to the Publication Stage. The Registration Stage involves a new Authentication and Key Agreement (AKA) protocol based on the ID-Based Cryptography (IBC) (Shamir, 1985). The IBC helps to certify the messages sender as the real owner of the NDO that will update the NetInf system. It also involves a capability-based access control mechanism (Gollmann, 2011) (Chen, 2014) to enforce authorized access to data objects. However, unlike the AKA protocol, the access control of (M. Aiash, 2014) has not been formally verified neither critically analysed. Therefore, this paper considers the security mechanisms in (M. Aiash, 2014), and supported by formal methods approach, we discover a security vulnerability in the proposed access control and introduce an enhancement. The enhance-

ment is formally verified using formal methods approach based on Casper/FDR tool.

The rest of the paper is organized as follows. Section 2 gives an overview of the NetInf architecture and the proposed Registration Stage in (M. Aiash, 2014). Section 3 formally verifies the proposed access control in (M. Aiash, 2014), describes the discovered attack and proposes an extension to address the discovered attack. The paper is concluded in Section 4.

2 RELATED WORK

2.1 An Overview of the NetInf

In NetInf architecture, publishers advertise data objects in the NetInf system and serve them to subscribers upon requests. The NetInf system acts as a middleman between publishers and subscribers, and is involved in configuring the forwarding path for data delivery (Edwall, 2013). Three pairs of messages have been defined as part of the NetInf architecture:

- The GET-REQ/GET-RESP messages: The GET message is used by a requester to request an NDO from the NetInf network. A node responding to the GET message would send a GET-RESP that is linked to the GET request using the message-Id (msg-id) from the GET message.
- The PUBLISH-REQ/PUBLISH-RESP messages: The PUBLISH message allows a publisher to push the name and a copy of the NDO to the network. A node receiving a PUBLISH message may choose to cache the NDO according to local policy and availability of resources and returns PUBLISH-RESP message, otherwise, it may choose to forward the message to other nodes without sending the response message.
- The SEARCH/SEARCH-RESP messages: The SEARCH message allows the requester to send a set of query tokens containing search keywords. The node that receives the SEARCH message, will either respond if the NDO is in its own cache or forward the SEARCH message.

These messages are supposed to be transported over a Convergence Layer (CL) protocol. As stated in (D. Kutscher, 2013), no CL protocol has been defined yet, but any protocol that allows NetInf messages to be passed without loss of information can be used as a NetInf Convergence Layer (NetInf-CL) protocol. These three pairs of messages define the transactions of the Publication and Data Retrieval Stages as follows:

1. **The Publish Stage:** Publishers publish their NDOs to the NetInf system by sending the PUBLISH-REQ message to the first hop node which might choose to cache the included information and responds with a PUBLISH-RESP message. Otherwise, it passes the PUBLISH-REQ to the next hop route. A node that caches NDO might update the NRS with the location of the NDO.
2. **The Data Retrieval Stage:** As shown in Fig 1, the NetInf combines two modes for data retrieval:
   (a) The Name Resolution: In this mode, the publisher publishes an NDO using PUBLISH message with a Name Resolution Service (NRS). In this case, a requester will approach the NRS first (using the GET message) which will direct him to the information publisher.
   (b) The Name-Based Routing: In this mode, the GET message will be forwarded hop-by-hop between NetInf nodes until a cached copy of the requested NDO is found or the original publisher is reached.

2.2 Verifying Security Protocols using Casper/FDR:

Previously, analysing security protocols used to go through two stages. Firstly, modelling the protocol using a theoretical notation or language such as the CSP (G. Lowe, 2009). Secondly, verifying the protocol using a model checker such as Failures-Divergence Refinement (FDR) (FDR, 1993). However, describing a system or a protocol using CSP is a quite difficult and error-prone task; therefore, Gavin Lowe has developed the CASPER/FDR tool to model security protocols, it accepts a simple and human-friendly input file that describes the system and compiles it into CSP code which is then checked using the FDR model checker. Casper/FDR has been used to model communication and security protocols as in (B. Donovan, 1999), (Aiash, 2014). The CASPER's input file that describes the systems consists of eight headers as explained in Table 1.

2.3 The NetInf Registration Stage

In NetInf, data sources publish NDOs by registering a name/locator binding with the NRS using the Publish message or announcing routing information in a routing protocol. Subscribers will approach the NRS requesting a specific NDO, and the NRS will first resolve the NDO into a set of available locators and then

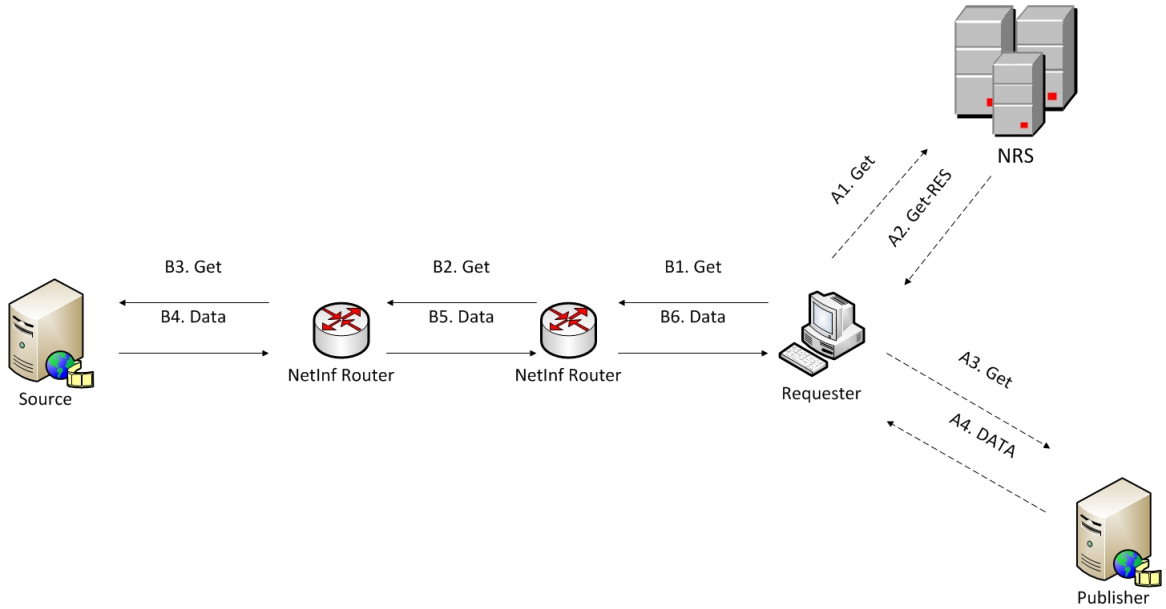

Figure 1: The NetInf Message Flow. The Name Resolution mode (dashed Arrows). The Name-Based Routing (solid Arrows) (M. Aiash, 2014)

Table 1: THE HEADERS OF CASPER'S INPUT FILE

The Header	Description
# Free Variables	Defines the agents, variables and functions in the protocol
# Processes	Represents each agent as a process
# Protocol Description	Shows all the messages exchanged between the agents
# Specification	Specifies the security properties to be checked
# Actual Variables	Defines the real variables, in the actual system to be checked
# Functions	Defines all the functions used in the protocol
# System	Lists the agents participating in the actual system with their parameters instantiated
# Intruder Information	Specifies the intruder's knowledge and capabilities

retrieve the copy of the data from the best available source.

In order to provide a secure data publication and retrieval, the authors in (M. Aiash, 2014) propose a new Registration Stage that comprises an Authentication and Key Agreement (AKA) protocol and authorization and access control mechanism. During the Registration Stage, both publishers and subscribers need to identify themselves to the NRS and acquire a security token that define their privileges and access rights. These tokens will be used to enforce authorized access to NDOs. The AKA protocol and an authorization mechanisms are briefly described in the following sections.

2.3.1 The Authentication and Key Agreement Protocol

The proposed Registration Stage involves a new Authentication and Key Agreement (AKA) protocol based on ID-Based Cryptography (IBC) scheme (Shamir, 1985). The scheme requires the presence of Trusted Key Generation (TKG) centres. However, unlike the normal Public Key Infrastructure (PKI) where a TKG randomly generates pairs of public/private keys, each node in IBC chooses its identifier (address or name) as a public key. Practically, any publicly known information that uniquely identifies the node could be used as a public key. The TKG generates the corresponding private key and securely distributes it to the node. More details about the AKA protocol including formal analysis and security discussion are found in (M. Aiash, 2014).

On the successful completion of the AKA protocol, mutual authentication between the party (Pub or Sub) and the NRS is achieved. The NRS will also be able to confirm the published NDOs as valid and authentic.

2.3.2 The Authorization Process Based on the Capability-Access Control

Once a party (subscriber or publishers) is authenticated, the NRS will generate security tokens. Two types of tokens are generated: Object Tokens (ObjToken), attached with the published NDOs and Subject Tokens (SubToken) attached with subscribers. These tokens define objects and subjects abilities. An ability is represented as a dot-separated sequence of numbers, called a label. So, an ability is a string $.i_1.i_2.i_3.....i_n$ for some value n where $i_1, i_2, i_3....., i_n$ are integers. Examples of abilities are .1.2.3, .4, or 10.0.0.5. Upon successful registration, both NDOs (objects) and subscribers (subjects) will be given labels (abilities) as parts of their security tokens. Access for an NDO is given if the NDO's label is a prefix of the subscriber's label. For instance, an NDO with a label ".3" could only be accessed by subscribers with abilities like ".3.1", ".3.2.3", ".3.1.2"...etc. This way, whenever an authenticated subscriber requests an NDO, he needs to present the right label that confirms his right to access the NDO.

Labels are generated by the NRS so subscribers can not promote themselves to access other NDOs, and they will be integrated into the security tokens. Furthermore, the security tokens are time stamped and have expiry date after which new tokens are needed. It also includes the party identification (ID), so a security token will have the following tuple: Token=ID, Label, Time Stamp, Expiry date. Using the time stamp and the expiry time will minimize the risk of a both active and passive replay attacks. Also, including the party identification (ID) will strive against impersonation attack. As part of the AKA protocol in (M. Aiash, 2014), the security token are distributed securely to the parties, and digitally signed by the NRS to guarantee token's integrity and authenticity.

3 The Authorized Data Retrieval Procedure

This section describes the transactions between subscribers, NRS and publishers to retrieve specific NDOs. It then describes the limitation of the authorization process described in section 2.3.2 and proposes enhancements.

3.1 The Authorized Data Retrieval Procedure

Based on the notations in Table 2. The procedure for accessing data goes as follows:

Table 2: Notation

The Notation	Definition
SK(X), PK(X)	The Private and Public keys of an entity X
Pub	The data source or the publisher of NDO
NRS	The Name Resolution Service which holds the name/location binding for NDOs
ObjToken	A security token will be attached to the published NDO
SubToken	A security token will be attached to the subscribers
h(m)	Hash value of the message (m)
{m}{K}	The message (m) being encrypted with the key (K)

After a successful authentication during the registration stage, the NRS generates the SubToken and passes it to the Subscriber as Msg1. The message is encrypted using the public key of the Sub, and the SubToken is hashed and digitally signed using the NRS private key for integrity and non-repudiation reasons. In Msg2, the Subscriber approaches the NRS to express interest of accessing a specific NDO using the Get packet of the NetInf. The NRS responds with the Get Response (GetRes) Packet as Msg3 which identifies the NDO's publisher.The message is encrypted using the subscriber's public key. As described in 2.3.1, the authentication protocol of the registration stage is based on ID-Based authentication scheme. In this scheme, nodes are free to choose any publicly known information to identify them and could be used as a public key. This eliminates the need for distributing the keys as in the traditional Public Key Infrastructure (PKI).

```
Msg1.  NRS→Sub:  {SubToken,
({h(SubToken)}{SK(NRS)}}{PK(Sub)}
Msg2.  Sub→ NRS: {Get}{PK(NRS)}
Msg3.  NRS→ Sub:  {GetRes}{PK(Sub)}
```

In Msg4, the subscriber approaches the publisher for accessing the NDO. The message includes the subscriber token and is encrypted using the PK(Pub). On receiving Msg4, the publisher approaches the NRS to check the validity of the included token(for conciseness these transactions are not shown below). If successful, the Pub compares the included SubToken against the object token (ObjToken). If the ObjToken is a prefix of the SubToken, access to the NDO is given and the publisher sends the requested

NDO as in Msg5.

```
Msg4.  Sub→Pub:{Get, SubToken,
({h(SubToken)}{SK(NRS)}}{PK(Pub)}
Msg5.  Pub→ Sub:  {NDO}{PK(Sub)}
```

3.2 Formal Analysis

To formally analyse the proposed solution, we simulate the system using Casper/FDR tool. The eight headings of the simulated system are described below.

The #Free Variables section defines the variables and functions that are used in the protocol. The term "Free Variables" refers to the fact that these variables will be represented by instances of actual values when running the protocol. For instance, the variables SubToken, ObjToken are of type SubjectToken and ObjectToken, respectively. The functions PK and SK return an agent's public key and private key, respectively. These functions will be defined later in the #Functions. The "InverseKeys" keyword defines the keys that are inverses of one another like PK and SK.

```
#Free variables
Pub, NRS, Sub :  Agent
SubToken:  SubjectToken
ObjToken:  ObjectToken
PK: Agent → PublicKey
SK: Agent → PrivateKey
InverseKeys = (PK,SK), (K1, K1),(K2, K2)
h :  HashFunction
```

The #Processes heading defines each involved agent in the protocol as a CSP process. The keyword "knows" defines the knowledge that the agent in question is expected to have at the beginning of the protocol run. In our system, INITIATOR, RESPONDER and SERVER are the names of the process representing the Subscriber, the Publisher and the NRS server, respectively. The values within the brackets and after the "knows" keyword define the agents' initial knowledge.

```
#Processes
INITIATOR(Sub, NRS, SubToken,Get, GetRes)
knows PK, SK(Sub)
RESPONDER(Pub, NRS, ObjToken, Data) knows
PK, SK(Pub)
SERVER(NRS, Pub, Sub, SubToken, ObjToken,
GetRes) knows PK, SK(NRS)
```

The #Protocol description heading defines the system and the transactions between the entities.

```
#Protocol description
0.  -> Pub :  NRS
1.  NRS -> Sub :{SubToken,
({h(SubToken)}{SK(NRS)}%TKO)%w}{PK(Sub)}
2.  Sub -> NRS: {Get}{PK(NRS)}
3.  NRS-> Sub:  {GetRes, Pub}{PK(Sub)}
4.  Sub -> Pub :  {Get, SubToken,
w%({h(SubToken)}{SK(NRS)}%TKO)}{PK(Pub)}
5.  Pub -> Sub :  {NDO}{PK(Sub)}
```

The security requirements of the system are defined under the # Specification heading. The lines starting with the keyword **Secret** define the secrecy properties of the protocol. The `Secret(Sub, SubToken, [Pub, NRS])` specifies the SubToken as a secret between the Sub, Pub and the NRS. The lines starting with **WeakAgreement** define the protocol's authenticity properties; for instance, the `WeakAgreement(Pub, Sub)` assertion could be interpreted as follows: if Pub has completed a run of the protocol with Sub, then Sub has previously been running the protocol, apparently with Pub.

```
#Specification
Secret(Pub, Data, [Sub])
Secret(Sub, SubToken, [Pub, NRS])
WeakAgreement(Pub, Sub)
WeakAgreement(Sub, Pub)
```

The # Intruder Information heading specifies the intruder identity, knowledge and capability. The first line identifies the intruder as Mallory, the intruder knowledge defines the Intruder's initial knowledge, i.e., we assume that the intruder knows the identity of the participants, all public keys, its own private key and can fabricate Register-Request and Register-Response messages.

```
#Intruder Information
Intruder = Mallory
IntruderKnowledge = {pub, nrs, Mallory,
PK, SK(Mallory),get, getRes}
```

After generating the CSP description of the systems using Casper and asking FDR to check the security specifications, the following attack has been discovered.

```
1a.  nrs -> I_sub :  {subToken,
{h(subToken)}{SK(nrs)}}{PK(sub)}
```

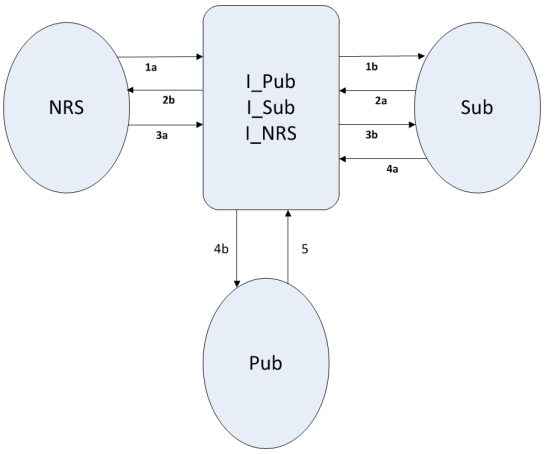

Figure 2: The Discovered MITM Attack

Figure 3: Casper/FDR Verification Result

1b. I_nrs -> sub : {subToken, {h(subToken)}{SK(nrs)}}{PK(sub)}
2a. sub -> I_nrs : {get}{PK(nrs)}
2b. I_sub -> nrs : {get}{PK(nrs)}
3a. nrs -> I_sub : {getRes, pub}{PK(sub)}
3b. I_nrs -> sub : {getRes, pub}{PK(sub)}
4a. sub -> I_pub : {get, subToken, {h(subToken)}{SK(nrs)}}{PK(pub)}
4b. I_nrs -> pub : {get, subToken, {h(subToken)}{SK(nrs)}}{PK(pub)}
5. pub -> I_nrs : {NDO}{PK(nrs)}

The notations I_sub, I_pub and I_nrs represent the case where the Intruder impersonates the Sub, Pub and NRS respectively. As shown in Fig 2, the discovered attack is a Man-in-the-Middle attack, where the Intruder intercepts and replays the messages. This attack could be interpreted as follows: the Sub believes (s)he is running the protocol, taking role INITIATOR, with the Pub, using data items subToken while it was with the intruder instead. Similarly, the Pub believes (s)he has completed a run of the protocol, taking role RESPONDER, with NRS, using data items NDO.

3.3 The Refined Version of the Proposed Mechanism

The problem with the initial version of the mechanism is that the Pub does not check if Msg4 is fresh and has been sent directly by the Pub. To accomplish such check, a challenge-response session between the Pub and the Sub will take place to guarantee the freshness of the message and the aliveness of the involved participants. The new sequence of messages looks as follows:

Msg1. NRS→Sub: {SubToken, ({h(SubToken)}{SK(NRS)}}{PK(Sub)}
Msg2. Sub→ NRS: {Get}{PK(NRS)}
Msg3. NRS→ Sub:{GetRes}{PK(Sub)}
Msg4. Sub→Pub:{Get, SubToken, ({h(SubToken)}{SK(NRS)}}{PK(Pub)}
Msg5. Pub→Sub: {n, SubToken}{PK(Sub)}
Msg6. Sub→Pub: {n}{PK(Pub)}
Msg7. Pub→Sub: {NDO}{PK(Sub)}

Unlike the initial version, the Pub will not send the requested NDO after receiving Msg4, rather it sends a randomly generated nonce (n) as challenge to the Sub in addition to the received SubToken as Msg5. On receiving this message, the Sub will verify that the included SubToken is the actual one sent in Msg4, hence it makes sure that Pub is alive. The Sub will return the challenge (n) as Msg6. This is a crucial transaction for the security of the proposed mechanism; an intruder will not be able to compose Msg6, it can only replay Msg5. The Pub will verify the included nonce in Msg6 and make sure that the Sub is alive. If everything is fine, the Pub will forward the requested NDO to Sub as Msg7. Furthermore, even if an intruder intercepts messages 1,2 and 3 and fabricates Msg4 with its own fabricated token, he will not be able to fake the signature of the NRS and hence the attack will fail on the check after Mag4. The new version of the mechanism has been checked with Casper/FDR and no attack has been found as shown in Fig 3.

It is worth to point out that the security of the proposed protocol is based on the system and the capability of the attacker as defined in (M. Aiash, 2014). Hence, the verification results might be different in

new system or with a more capable attacker. Furthermore, it functionalities of the system entries such as the NRS node has been defined as part of the proposal in (M. Aiash, 2014). The authors acknowledge that different design might scale and prove to be more resilient, however, such modifications to the system might increase the complexity of the system in terms of establishing a trust relationship between the new entity and the existing ones.

4 CONCLUSIONS

Network of Information (NetInf) is one proposed approach for Information-Centric Networking (ICN). In NetInf, publishers publish their data (through the Publication stage) to the NRS system which then launch these data to subscribers upon request (through the Data Retrieval Stage). Previous work in the literature has explained how the Publication Stage is vulnerable to masquerading and content poisoning attacks which might happen when an unauthenticated node publishes invalid data to the system. The work proposed authentication protocol is based on the IBC protocol and achieves mutual authentication between publishers and the NetInf system and enforce authorization. This paper discovers a security flaw in the access control and authorization mechanism and using formal methods approach propose enhancements to address the security flaw.

REFERENCES

(1993). *Failures-divergence refinement: fdr2 user manual and tu- torial*.

Aiash, M. (2014). A formal analysis of authentication protocols for mobile devices in next generation networks. *Concurrency and Computation: Practice and Experience*.

B. Ahlgren, C. Dannewitz, C. I. D. K. B. O. (2012). A survey of information-centric networking. *IEEE Communication Magazine*.

B. Donovan, P. Norris, G. L. (1999). Analyzing a library of security protocols using casper and fdr,. In *In Workshop on Formal Methods and Security Protocols*.

Chen, H. C. (2014). A multi-issued tag key agreement with time constraint for homeland defense sub-department in nfc environment,. *Journal of Network and Computer Applications*, pages 88–98.

D. Kutscher, S. Farrell, E. D. (2013). The netinf protocol,. Technical report, Internet Draft.

Edwall, T. (2013). The network of information: Architecture and applications. Technical report, SAIL Project.

G. Lowe, P. Broadfoot, C. D. M. H. (2009). Casper: A compiler for the analysis of security protocols. Technical report, Oxford.

Gollmann, D. (2011). *Computer Security*. Wiley,, London, 2rd edition.

H. Baker, R. Stradling, S. F. D. K. B. O. (2012). The named information (ni) uri scheme: Optional features. Technical report, Network Working Group.

J. Loo, M. A. (2014). Challenges and solutions for secure information centric networks: A case study of the netinf architecture. *Journal of Network and Computer Applications*, 50:6472.

K. Pentikousis, B. Ohlman, E. D. S. S. G. B. P. M. (2013). Information-centric networking: Evaluation methodology,. Technical report, Internet Draft.

M. Aiash, J. L. (2014). An integrated authentication and authorization approach for the network of information architecture. *Journal of Network and Computer Applications*, 50:7379.

Shamir, A. (1985). Identity-based cryptosystems and signature schemes,. In *CRYPTO 84 on Advances in cryptology,*. Springer-Verlag.